

Straight ~ Talking

The RS232 lead is so infamous, it even got sung about on *Spitting Image*. But despite the continuing bad PR for RS232 and serial communications in general, I'm convinced that most implementations are standard and with a little understanding there's nothing to fear.

There are two types of device that are traditionally attached to each other by serial connections. One kind is referred to as 'data communications equipment', or DCE (characteristically a modem) and the other as 'data terminal equipment', or DTE (characteristically a computer terminal). As you can see, it's the middle initial where the distinction arises – all you have to remember is C for comms and T for terminal.

The other basic concept needed to comprehend the subject is the historic purpose of an RS 232 serial link, from which everything else logically follows. Picture two VDU terminals connected via a phone line. The first VDU plugs into a modem, which attaches to the phone line, to which is attached the other modem, which in turn is plugged into the second VDU. The arrangement is symmetrical: whichever end you start from, the sequence of connections is the same.

Now to the specifics. The lead from the VDU to the modem happens to have 25 wires and electrical characteristics that are defined by the RS232 specification. To keep things easy, each pin on the VDU is connected to exactly the same numbered pin in the modem. Since the VDU has a male connector and the modem a female one, if you got the two pieces of equipment close enough together you wouldn't even need the cable – one would plug straight into the other.

There's also a 9-pin variant of the serial connection, but we'll come to that later.

Let's look first at the data signals (see Figure 1). These are named from the point of view of the phone line. Pin 2 carries the 'transmit data' from the terminal to the modem, and pin 3 returns the 'received data' from the modem to the terminal. Pin 7 is a common ground, which simply fulfills that electrical function.

Note that Pin 2 has the same name whichever end of the cable you look at: it's always called 'transmit data', even though at the modem end it's an input to the equipment rather than an output.

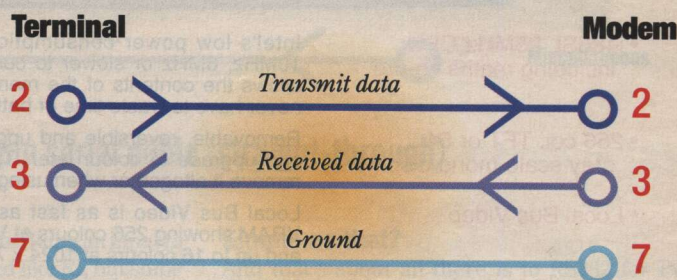
Pins and needles

Now, what would happen if we wanted to connect a terminal straight to another terminal, without two modems and a phone line in the way? This time we can't just connect pin 2 to pin 2 and pin 3 to pin 3, because the pin 2s are both outputs and the pin 3s are both inputs. We have to connect the output from one terminal to the input of the other, and vice versa.

This is simply done by swapping the connections over, giving a cable where the wires between pins 2 and 3 are crossed over (see Figure 2). This type of cable, which is still symmetrical – it doesn't matter which way round it's plugged in – has a name: a 'null modem cable'. As you can now see, it's so called because it takes the place of (nullifies) the modems in the link between the two terminals.

You may be wondering about all the other things with RS232 interfaces that aren't exactly terminals or modems. In the world of IBM PCs, the rule is simple: if it isn't a modem (and by that I mean a real modem attached to a phone line) it has to be a terminal. So as well as the PC itself, which is obviously a data terminal, a peripheral such as a serial printer is classified – and wired – as a terminal. That means it ought to have male connectors, but in fact most suppliers of printers fit them with female connectors – probably for electrical safety reasons. Even so, a printer's pinout will always be that of a terminal, so a null modem cable is required to connect a printer to a computer (both of them being terminals). You mustn't use a simple straight-through serial lead, or the system won't work – for reasons which should now be glaringly obvious.

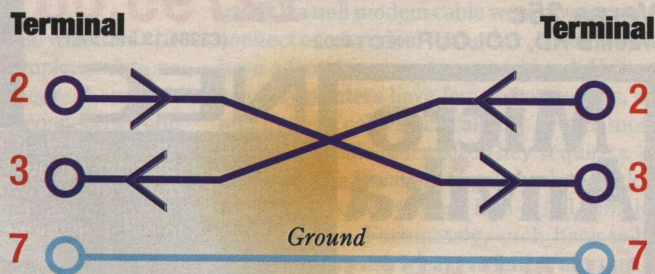
The notorious Roland Perry ponders a leading question and makes a significant connection



1 Data signal pinout of 25-pin serial cable

There's one exception to this simple rule, and that's the serial mouse. Because it plugs directly into the serial port on a computer, the female connector on the end of the mouse lead is wired like the connector on the back of a modem. So it's DCE, not DTE. But then you won't need to know that anyway, because the mouse will be supplied with its own lead.

Serial mice usually have 9-pin connectors. There's nothing very special about a 9-pin RS232 connector – it just has a few pins renumbered. A normal 25-pin RS232 connector doesn't use pins 9 to 19 anyway. The table below shows the correspondence between 25 and 9-pin cables; adaptors from one to the other are readily available. Beware: some adaptors don't connect all nine pins, as they're intended to connect 9-pin mice to 25-pin ports; most mice only use pins 2, 3, 4, 5 and 9.

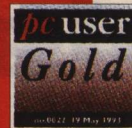


2 Data signal pinout of null modem cable

Signal	25-pin	9-pin
Frame ground	1	NA
Transmit data	2	3
Received data	3	2
Request to send (RTS)	4	7
Clear to send (CTS)	5	8
Data set ready (DSR)	6	6
Signal ground	7	5
Data carrier detect (DCD)	8	1
Data terminal ready (DTR)	20	4
Ring indicator (RI)	22	9

THE ULTIMATE **VERSA**-TILITY

A powerful mobile office with easy upgrade capability and a wide range of options



- i486SL 25MHz CPU including maths co-pro
- 256 col. TFT or 64 gray scale mono display
- Local Bus Video
- Up to 12 hours battery life with NEC's DoubleTime Technology
- 80/120MB removable hard drive
- 4MB RAM •
- 2 PCMCIA Type II slots Supports ExCA (Exchangeable Card Architecture)
- Enhanced Parallel Port

Intel's low power consumption CPU can be switched from 25MHz to 10MHz, 6MHz or slower to conserve battery power. Phoenix Miser BIOS allows the contents of the memory to be saved to the hard disk. You will never have to waste time or battery power rebooting!

Removable, reversible and upgradeable display. You can start with mono and upgrade to colour later. Turn the display around for presentations or remove it altogether when using an external monitor.

Local Bus Video is as fast as the fastest desktop machines, with 1MB VRAM showing 256 colours at VGA; 800 x 600, 256 colours and up to 16 colours at 1024 x 768 on an external monitor.

NiMH batteries each give up to 5 hours operation with colour display and up to 6 hours with mono display. The floppy disk drive can be removed and replaced with a second battery to operate for a full working day without recharging!. A 5 minute bridge battery allows the batteries to be removed without switching off.

The 120 MB drive can be removed for security or to upgrade to a larger drive. Additional 120MB and 180MB drives are available now with 240MB coming soon.

Upgradeable to 8MB, 12MB or 20MB; or beyond using PCMCIA RAM cards.

Accepts either 2 Type II PCMCIA cards such as Flash ROM cards up to 20MB, RAM cards, Fax modems or network cards; Or a single Type III device such as a hard disk drive. ExCA allows the user to swap cards without powering down the machine.

Outputs data at 30 times standard parallel port speed for faster communication with laser printers or parallel to SCSI adaptors or parallel port network adaptors.

PORTABLE CD-ROM

NEC CD R38 £275.00
NEC51221 (£323.00inc.VAT)



UltraLite Versa 20
486, 20MHz, 80MB HD, MONO NEC 14403

£1399.00
(£1643.83 inc. VAT)

UltraLite Versa 25
486, 25MHz, 120MB HD, MONO NEC 14401

£1799.00
(£2113.83 inc. VAT)

UltraLite Versa 25c
486, 25MHz, 120MB HD, COLOUR NEC 14402

£2795.00
(£3284.13 inc. VAT)



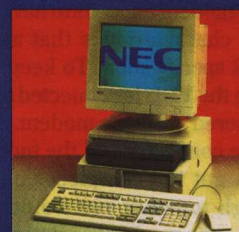
• **UltraLite Versa 25 •**
4MB RAM, 120MB hard drive as standard, 9.4" mono LCD super twist nematic screen. Serial port, parallel port, Video port, Keyboard interface, Mouse interface, Docking interface, PCMCIA expansion (1 of type III, or 2 of type II)



• **TRULY MODULAR •**
RAM, hard drive and Liquid Crystal Display are all easy to upgrade yourself.

OPTIONS AVAILABLE...

Docking Station
Perfect Touch UK keyboard
MultiSync 3FG monitor
MultiSync 4FG monitor
Colour TFT LCD upgrade
4MB memory upgrade
8MB memory upgrade
16MB memory upgrade
180MB hard disk upgrade
240MB hard disk upgrade
NiMH battery pack
AC adaptor
Car DC adaptor
Carry case



• **DOCKING STATION •**
Convert the UltraLite Versa into a powerful full feature desktop workstation. The Docking Station can be used by a number of different users.

Docking Station
£399.00
(£468.83 inc.VAT)
NEC19501

Call us and order now on
071 636 2547

FAX

071 436 9471



Goods subject to availability and prices subject to change without notice. E&OE

**Micro
Anvika**
C.o.m.p.u.t.e.r
solutions

NEC

Please come in and visit our showrooms. Apart from these outstanding products we stock a wide range of computer hardware, networks and software, including books and accessories as well as peripherals and consumables to meet all your computing needs

Opening hours:

Monday to Friday 8:30am to 6:00pm
Saturday 9:30am to 6:00pm

53-54 Tottenham Court Road London W1P 9RE
251 Tottenham Court Road London W1P 9AD
13 Chenies Street London WC1E 7ET

All prices subject to VAT
@ 17.5%

Spotting the signals

Besides the transmit and received data, there are a lot of other signals (see Figure 3). What can they possibly be doing? Most of them are either 'connection established' or flow control signals. Once again, the ancestry of these can be traced to the original terminal-modem-phone-modem-terminal arrangement. Before data can be exchanged, it's necessary for both the terminal and the modem to signal, "I'm ready and prepared to exchange data!"; then, while the data is flowing, there has to be a means for the receiving end to say "Hang on a minute!" if the rate of sending is more than it can cope with.

Let's look first at the data flow from the terminal to the modem. To begin with, the terminal will give the signal 'data terminal ready' (DTR); when the modem is successfully connected to another over the phone line, it will reply with 'data set ready' (DSR). The modem will perform flow control through the 'clear to send' (CTS) line, which the terminal might expect to see changing state frequently; whereas the DSR signal should remain steady as long as the modem is powered up and online.

When the terminal releases DTR, the modem should hang up the phone and drop DSR. Meanwhile, the terminal (which will, of course, normally be a PC) can perform its own flow control on the data arriving from the modem by lowering 'request to send' (RTS) when it wants to stem the tide of data.

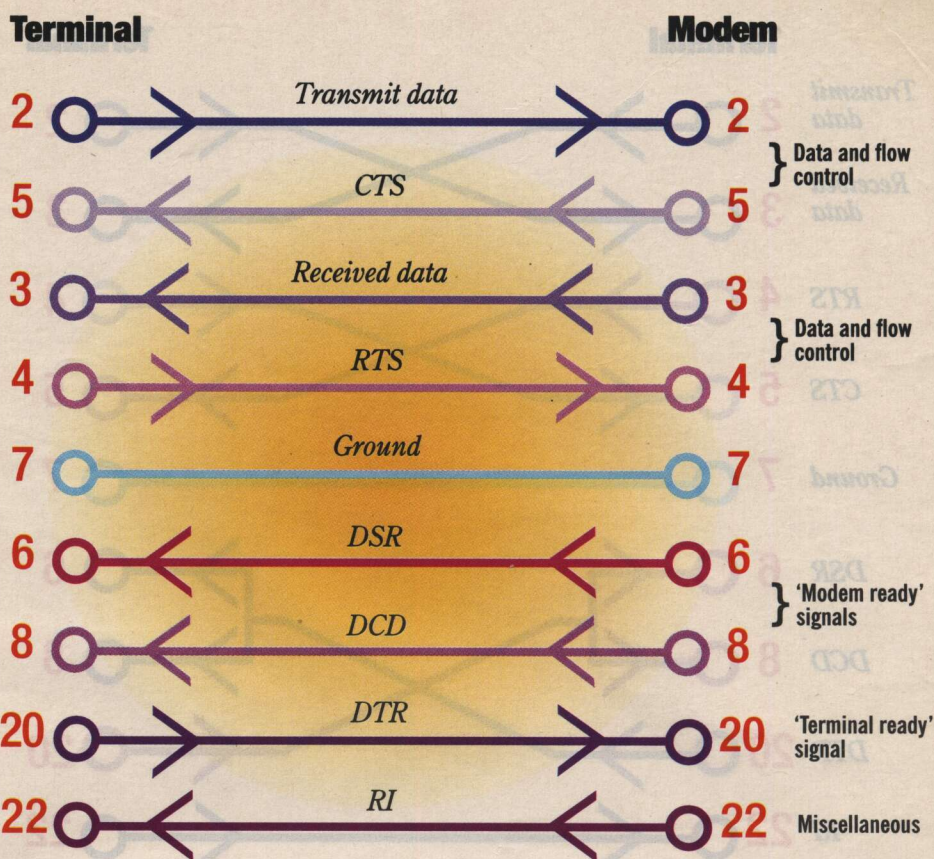
Two further signals are available from the modem: 'ring indicator' (RI) and 'data carrier detect' (DCD). These can be used to tell if the incoming phone line to the modem is ringing, and if the modem is locked on to the whistling tones that carry the data.

So far, so good. The really tricky part comes when we need to make a null modem cable that takes into account the control signals as well as the data. A really strict interpretation says that the "I'm ready!" signals should simply be looped back at each end, because the cable is replacing a pair of modems and is always 'ready', as long as it's plugged in. This would mean that as soon as one end said "I'm ready!", it would hear the same signal echoed back, as if from the other end, and therefore start sending – which is what we want.

In practice, however, the way many PCs and communications software use the various status lines means that the two "I'm ready" lines should be passed through to the other end. What this means is that the DTR output signal from one end of the cable should be connected to the DSR input at the other – and vice versa. Similarly, the RTS and CTS lines need to be crossed.

Cross purposes

It isn't 100 percent clear what to do with the DCD and RI lines. Most null modem implementations leave RI disconnected. In theory, though, it ought to be possible for a user to connect two null modem cables in series and thus make a straight-through serial cable (the two crossovers cancelling each other out), and on that basis I recommend connecting RI straight through. RI is an input to the terminal, and no harm should come to connecting two inputs together at either end if the null modem cable is used to connect two terminals together.



3 Full pinout of 25-pin serial cable (straight-through)

DCD is a thornier problem. Being an input to the terminal it could be treated like RI, but some printers require it to be asserted before they'll print, and the DCD input to the computer isn't going to satisfy that. For that situation DCD needs to be tied to an output, with DTR at the other end of the cable being the favourite choice.

However, if wired like that, this null modem cable, when connected to a modem, will short the DCD and DSR outputs of the modem together. It may sound unlikely that you'd want to connect a null modem cable to a modem (rather than between two terminals), but a null modem cable can just as easily be used to connect two devices that are socketed as modems – for example, a modem and the connectivity cable provided for the HP 100LX palmtop computer. Nevertheless, there seems little alternative to taking this risk, without creating as serious a side-effect somewhere else. The null modem cable pinout that results from these decisions is shown in Figure 4 (over page).

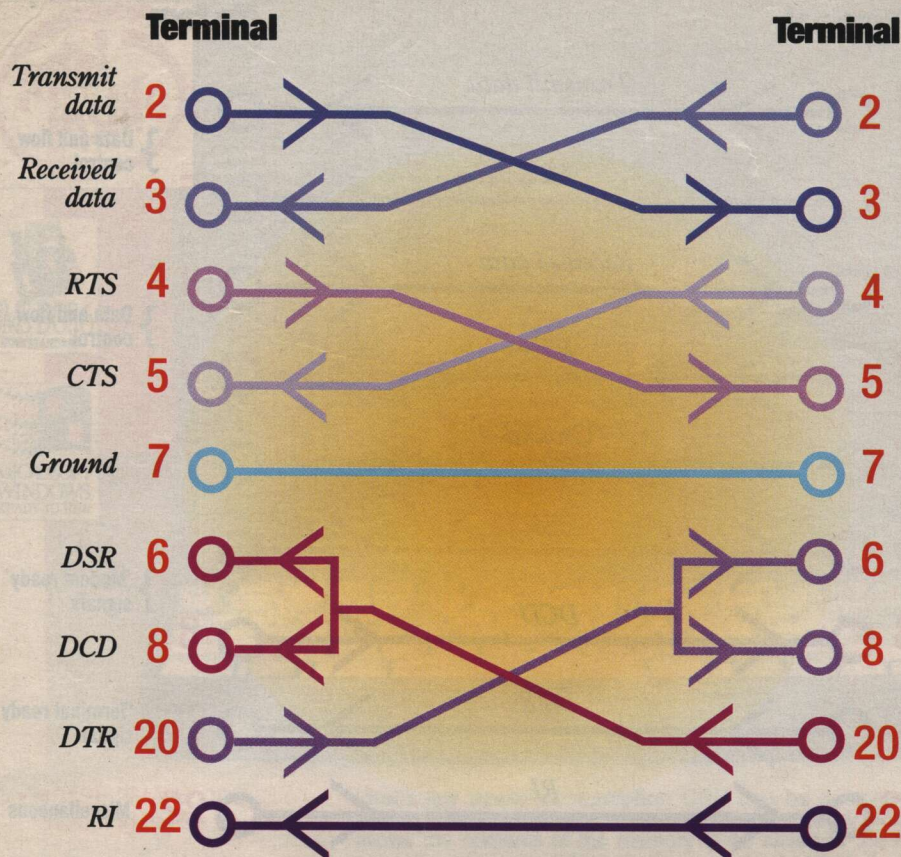
What about the physical connector style used? If a null modem cable were always used to connect one terminal to another, it would always have a female connector at each end. However, as many printers have female connectors, most null modem cables, and almost all null modem boxes, are male-to-female. They require a 'gender bender', a female-to-female back-to-back connector, to convert them into a full female-to-female cable. Fortunately, such back-to-back connectors are fairly easy to find. Unfortunately, the converse, a male-to-male back-to-back connector for null modem cables connecting two modems, is much more difficult to get hold of.

Simple as that?

And that's about all there is to RS232 for PC users. Now, I know a few purists and mainframe users will be jumping up and pointing out that 'real' computers are wired as modems (so that terminals can plug straight into them) and that a null modem cable really ought to swap DTR/CTS and RTS/DCD (not RTS/CTS and DTR/DSR), even though that leaves DSR in the same limbo as discussed for DCD. From such debates flow the confusion that has blemished RS232's reputation. What I've detailed here is what will work for most PC users, and as far as we at *Shopper* are concerned, that's what matters!

Despite the simplicity of the serial connection itself, however, it can't be denied that serial communications is one of the most troublesome subjects yet invented to frustrate the PC user. Not only is it technically fraught, but it also has a reputation for a lack of standardisation. Even new operating environments like *Windows* appear to do little for the plight of communications users. If anything, they make it worse. So what's the problem, and how can it be overcome?

Modems are one of the most common serial peripherals, and much of the trouble is the fault of modem makers, who build ever more divergent command languages into their equipment while continuing to claim a core of 'Hayes compatibility'. As modems get more sophisticated, new features – such as error correction and data compression – need to be configured as circumstances dictate. And rather than have front-panel switches or their own user interfaces, modems demand that impenetrable commands like &D1 be sent over the main serial interface.



4 Full pinout of 25-pin null modem cable

Hayes has a patent on the way of swapping to command mode (a string of three '+' signs with a one-second gap before and after), and this is standard across almost all modems, as are simple commands like switching between tone and pulse dialing. The setup problems arise from variations in the more complex commands which many users don't even know they need to understand. When all that's sorted out the problems still aren't over, because the speeds at which modern modems work can overwhelm even relatively high-powered PCs if they're fitted with standard serial hardware and using standard software drivers.

Playing squash

Data compression efficiency is often overstated, like that of disk compression software. The two work in similar ways, both transparently to the user. Normal text files, like the words of this article, will probably compress about 1.8 to 1 on disk (that is, the size of a file will reduce by 45 percent) and by 2.2:1 over a modem with so-called 'V42bis' compression. Claims of compression ratios of up to 4:1 with V42bis would only be realised with extremely repetitive data.

Even when the modem provides data compression, the computer must be capable of accepting the data as it is, uncompressed, at a rate at least double the underlying operating rate. For example, a 14.4Kbps (thousand bits per second) modem would require at least a 38.4Kbps serial link. The next lower speed, 19.2kbps, wouldn't be adequate, and might even be a squeeze for a modem working at 9.6Kbps.

The original IBM PC's serial port used an 8250 Uart, and the AT (the standard on which

most PC-compatibles have been based) a 16450. These are the devices that transmit and receive the serial data, and they have a one-byte-wide interface to the PC. Every byte has to be handled separately, and when the PC is multitasking – eg under *Windows* – it's possible for a byte to be missed if another is received before the Uart has been emptied of the previous one. As a result, *Windows* doesn't even attempt to offer rates over 19.2Kbps. Network traffic, disk accesses and, above all, *SMARTDrive*'s write cache contribute to periods of 'deafness' during which high-speed serial communications can lose characters.

The most effective solution is to replace the Uart with a more sophisticated one, the 16550. Because this contains a 16-byte memory of its own (called a Fifo – 'first in, first out' – system), the PC has much longer to respond to received data. If it's set up to trigger when eight characters have been received, it then has a further eight character times to fetch, and store elsewhere, the handful of bytes that have arrived. This enhanced Uart requires adequate software support, which is built into some Dos comms packages and, to an extent, into *Windows 3.1*.

Since the days of *Windows 3.0*, however, I've been using a far superior set of drivers, called TurboCom, which provide some significant improvements over anything from Microsoft. As well as providing support for non-standard port addresses and resolving various issues to do with interrupt sharing, TurboCom is more efficient. The clincher for me is the way it provides a virtual 16550 to a 16550-aware comms program running in a Dos box under *Windows*. Previously the *Windows* comms driver would only offer a virtual

8250 even if a 16550 was fitted, so the buffering of the 16550 was lost. TurboCom goes one stage further too, providing its buffering to Dos programs that *aren't* 16550-aware.

Some systems aren't geared up to having a 16550 installed. To fit one requires being able to remove the former Uart from its 40-pin socket. More and more PCs are built with the functionality of the Uarts integrated into a larger chip. Card modems, either the traditional sort or PCMCIA, have a Uart in them, but once again this is more than likely to be inaccessible. Integrated modem and Uart chips to make high-speed PCMCIA modems with built-in 16550 are not yet available from the suppliers. However, if your PC has a free card slot, all is not lost: flexible add-in cards to fit one or two 16550s are readily available.

No visible means of support

ONE THING THAT ISN'T READILY AVAILABLE, in the PC stores that I visit at least, is much support for the 9-pin version of RS232. Even a simple extension cable is very hard to find, and the part I've been searching for recently – a male-to-male gender changer – is remarkably elusive.

Software support is something else I've been having a lot of trouble with of late. A month after my excursion into the world of serial comms with the HP100LX review, one of my questions has still to be satisfactorily resolved. H-P Europe has failed to explain why the MSDos 6 *Interlink* utility doesn't work with it, although the word from the HPHANDHELD forum on CompuServe is that H-P doesn't care because the HP100 Connectivity Pack has the same function. Unfortunately, the Connectivity Pack still isn't available.

The Microsoft people on the MSDOS forum haven't got back to me about the message shown when *Interlink*'s remote boot fails, but I've found *LapLink 3*'s remote boot does work. Remote boot allows the file copying software to transfer itself to the remote PC even though there's no comms software yet installed on that machine.

There are signs that Microsoft's cool support for serial comms is beginning to thaw. The inclusion of *Interlink* in the Mobile Computing version of MSDos 5 and all versions of 6 was an indication that it wanted to get into the business of serial (and parallel) port networking. Until then, the market for cheap PC-to-PC data transfer was dominated by Traveling Software's *LapLink* and *DeskLink* and similar offerings from East Coast called *WinLink* and *Lap2Desk*.

At the launch of *Windows for Workgroups*, Microsoft mentioned there'd be a serial port driver for occasionally connected PCs, and there was a suggestion that a PC could dial into the network from outside using a modem. I have yet to see that driver, but the next version of *WfW* is expected to allow sharing of modems in the same way it allows sharing of disks and printers. Proper modem sharing, currently an elusive facility, will surely boost the take-up of all kinds of online systems. ■

Contacts

Dynamic Communications Systems
(0272) 255465

Traveling Software
(0753) 831855

East Coast Software
(Eire 010 3531) 283 1166